10Б

УДК 044.93.2

Березький О.М., Батько Ю.М.
Тернопільський національний економічний університет, м. Тернопіль, Україна
ob@tneu.edu.ua, prog_net@ukr.net

Алгоритм проходження контуром об’єкта з використанням зворотного ходу

У статті представлено аналітичний огляд сучасних алгоритмів проходження контуром, визначено їх переваги та недоліки. Запропоновано удосконалений алгоритм проходження контуром об’єктів з використанням зворотного ходу «Backward contour tracing». Розроблений алгоритм протестовано на цитологічних зображеннях.

Вступ

В багатьох задачах обробки зображень аналіз форм об’єктів є актуальним і інформативним. При аналізі біомедичних зображень важливу роль відіграє дослідження форми мікрооб’єктів [1-4]. Широкого розповсюдження контурний аналіз отримав при обробці цитологічних, гістологічних, томографічних, рентгенівських зображень, тобто в тих випадках, коли структуру та колір визначити набагато складніше, ніж контур. Сьогодні контурний аналіз широко використовується в топографії для дослідження карт [5], в архітектурі для побудови 3D моделей [6] тощо. Оскільки основна інформація про форму об’єкта міститься в контурі об’єкта, то виділення та опис контуру є важливою задачею аналізу зображень.

Як відомо, контур об’єкта – це частина об’єкта, яка містить велику кількість інформації про форму об’єкта та слабо залежить від кольору та текстури зображення [7].

Для визначення контуру використовують такі підходи: підкреслення різких перепадів яскравості або виділення однорідних областей з подальшим проходженням контуром.

До алгоритмів першого типу слід віднести: алгоритм змії, алгоритм Кенні, фільтрацію на основі операторів Собеля, Лапласа, Превіта тощо [8-12]. Дані алгоритми базуються на підкресленні різких перепадів яскравості, які характерні краям об’єктів. Результатом їх роботи є набір незв’язних областей. Для отримання зв’язного контуру необхідно провести додаткову обробку, наприклад використати алгоритми морфологічного аналізу.

Серед відомих алгоритмів виділення областей є такі: порогова сегментація, кластеризація, нарощування областей, алгоритм водоподілу, блочна сегментація тощо [8-12]. Дані алгоритми базуються на об’єднанні пікселів в однорідні області на основі певного закону однорідності або ознаки. Результатом їх роботи є набір однорідних областей. Для отримання зв’язного контуру об’єкта необхідно використати алгоритми проходження контуром. Проходження контуром – процес послідовного перебору пікселів цифрового зображення за певними правилами з метою знаходження зв’язного контуру об’єкта.

Уведемо визначення, які будуть необхідні для розуміння розробленого алгоритму.

Стартовий піксел Ps – піксел, з якого починається обхід контуром об’єкта. Вибір стартової точки проводиться довільно, наприклад крайній лівий верхній піксел, що належить об’єкту.

Кінцевий піксел *Pe* – піксел, потрапляючи на який алгоритм завершує свою роботу.

Активний піксел *Pa* – піксел, що знаходиться в середині розміточної сітки.

Сусідній піксел *Pn* – піксел, який межує з активним пікселом.

Контурний піксел *Pc* – піксел, що належить контуру об'єкта.

Фоновий піксел *Pf* – піксел, що належить фону зображення.

Сусідній контурний піксел *Pcn* – піксел, що належить контуру об'єкта та межує з активним пікселом.

Розміточна сітка – таблиця, що служить для вибору послідовності перевірки сусідніх пікселів (рис. 1).

8	1	2
7	*Pa*	3
6	5	4

Рисунок 1 – Приклад розміточної сітки

На сьогоднішній день відомі декілька алгоритмів проходження контуром:

1. Алгоритм «жука» (Square Tracing Algorithm) [11]. Основна перевага даного алгоритму – простота. Проходження контуром здійснюється на основі двох простих правил: якщо значення активного піксела дорівнює одиниці (активний піксел належить однорідній області), то проводиться поворот ліворуч; якщо значення активного піксела дорівнює нулю (активний піксел не належить однорідній області), то проводиться поворот праворуч. Алгоритм зупиняє свою роботу, якщо він повернувся в стартовий піксел.

2. Алгоритм «Moore-Neighbor Tracing» [8] базується на покроковій перевірці всіх сусідніх пікселів з метою знаходження наступного контурного піксела. Пошук наступного контурного піксела починається з піксела, з якого відбувся перехід на активний піксел. Зупинка роботи алгоритму відбувається при поверненні алгоритму в стартовий піксел.

3. «Redial Sweep» [9]. Даний алгоритм є модифікацією попереднього. Основна відмінність полягає у виборі піксела, з якого починається пошук наступного контурного піксела. У даному алгоритмі це піксел, який був визнаний контурним на попередньому кроці алгоритму, а не піксел, з якого відбувся перехід на активний піксел.

4. «Theo Pavlidi's Algorithm» [12]. Основна ідея даного алгоритму полягає у використанні групи з трьох пікселів для визначення наступного контурного піксела (рис. 2). Перевірка здійснюєтсья за наступними правилами: спочатку піксел P1 (згідно з рухом стрілки, рис. 2), якщо піксел P1 не є контурним, то проводиться перевірка піксела P2, при негативному результаті перших двох пікселів – перевіряється піксел P3. Якщо жоден з пікселів не є контурним, то проводиться поворот на кут 90 градусів за годинниковою стрілкою, в іншому випадку поворот здійснюється згідно зі схемою поворотів (рис. 2).

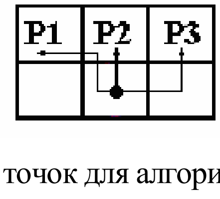


Рисунок 2 – Приклад групи з трьох точок для алгоритму «Theo Pavlidi's Algorithm»

На теперішній час дані алгоритми широко використовуються для проходження контуру [13-16]. Практично для кожної мови програмування існують бібліотеки з реалізаціями даних алгоритмів.

Основним недоліком наведених алгоритмів є залежність від складності контуру та критерію зупинки. Алгоритми надзвичайно чуттєві до об’єктів, контур яких містить відгалуження товщиною в один піксел. Дана ситуація може призвести як до помилкового завершення роботи алгоритмів, так і до некоректного виділення контуру. Аналогічна проблема може виникнути, якщо об’єкт складається з двох і більше частин, що з’єднані між собою тільки одиночними пікселами. Інший недолік алгоритмів полягає у недосконалих критеріях зупинки (повернення в стартову точку, проходження точки n раз). Для складних контурів дані критерії зупинки призводять до некоректних результатів роботи. Тому розробка алгоритму проходження контуру є актуальною задачею.

Алгоритм «Backward contour tracing»

Проаналізувавши переваги та недоліки існуючих алгоритмів, запропоновано наступний алгоритм проходження контуром з можливістю зворотного ходу («Backward contour tracing»):

1. Проводиться пошук стартового піксела Pa .
2. Проводиться пошук сусіднього контурного піксела Pn за годинниковою стрілкою.
3. Проводиться пошук сусіднього контурного піксела $P'n$ проти годинникової стрілки.
4. Якщо отримані сусідні контурні піксели в п. 2 та в п. 3 співпадають, $Pn = P'n$, то активний піксел визнається як фоновий Pf і виключається з подальшої обробки і здійснюється перехід до п. 8.
5. Якщо отримані сусідні контурні піксели не співпадають $Pn \neq P'n$, то стартовий піксел визнається також і кінцевим $Ps = Pe$. Сусідній контурний піксел, отриманий на кроці 2, заноситься в масив контурних пікселів і йому присвоюється мітка активного піксела $Pa = Pcn$.
6. Проводиться пошук наступного сусіднього контурного піксела Pcn . Послідовність перевірок сусідніх контурних точок відбувається на основі розміточної сітки. Піксели перевіряються за годинниковою стрілкою. Позиція стартової перевірки d визначається як $(d'+2)\bmod 8$, де d' – позиція, з якої було знайдено активний піксел Pa . Пошук завершується при знаходженні наступного контурного піксела або до перевірки всіх сусідніх контурних пікселів.
7. Якщо сусідній піксел є контурним та не співпадає з кінцевим пікселом, то він заноситься в масив контурних пікселів і йому присвоюється мітка активного піксела $Pa = Pcn$ та проходить перехід до п. 6.
8. Якщо знайдений сусідній піксел розмічений на попередніх кроках пошуку, але не співпадає з кінцевим пікселом $Pcn \neq Pe$, то активний піксел визнається малоінформативним, видаляється з масиву контурних точок, індексується як піксел фону. Статус активного піксела присвоюється попередньому контурному пікселу.
9. Якщо знайдений контурний піксел співпадає з кінцевим пікселом $Pcn = Pe$ та кількість точок, що належать контуру, більша за 1, то алгоритм завершує роботу.

Результатом роботи алгоритму є масив контурних точок (рис. 3). На рис. 3 а) проілюстровано процес знаходження двох сусідніх контурних пікселів P та D, при цьому було відкинуто піксели A, B, C, оскільки вони є сусідніми відносно піксела P, проте не є контурними. На рис. 3 b) наведено покрокове проходження контуру об’єкта алгоритмом «Backward contour tracing», світлими стрілками показано перевірки, які мали негативний результат, чорними – позитивний. При коректному завершенні роботи алгоритму отримуємо масив точок $\{A_1, A_2, \ldots, A_m\}$, що утворюють контур об’єкта або повідомлення про помилку, при негативному результаті.

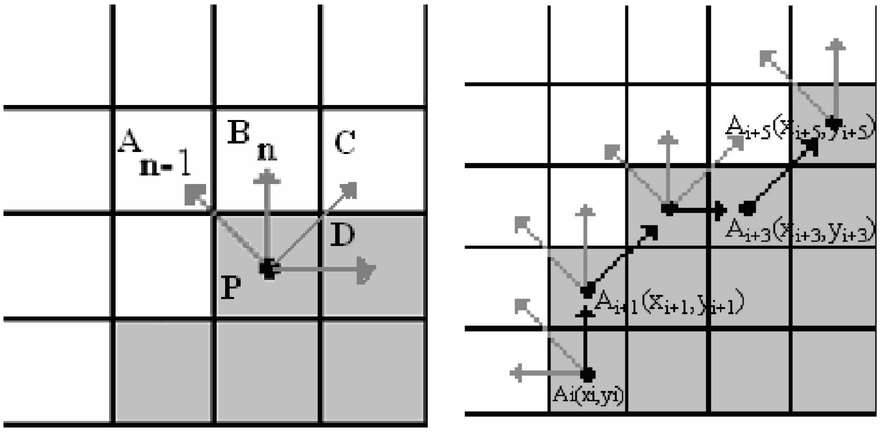
До переваг даного алгоритму слід віднести:

– роботу з 8-зв'язними контурами;

– незалежність від вибору початкового піксела;

– високу швидкодію за рахунок зменшення точок для аналізу, що, наприклад, в порівнянні з алгоритмом «Redial Sweep» може скласти до 25%;

– можливість відкоту роботи. Необхідно зазначити, що принцип відкоту роботи алгоритму проходження контуром раніше не використовувався. Використання відкоту роботи дозволяє уникати зациклення алгоритму, відкидання малоінформативних пікселів та можливість коректної роботи зі складними контурами.



a) b)

Рисунок 3 – Приклади роботи алгоритму «Backward contour tracing»: a) – приклад вибору наступного піксела для перевірки; b) – приклад перебору проходженням контуру алгоритмом «Backward contour tracing» (чорні стрілки – позитивні перевірки, світлі – негативні)

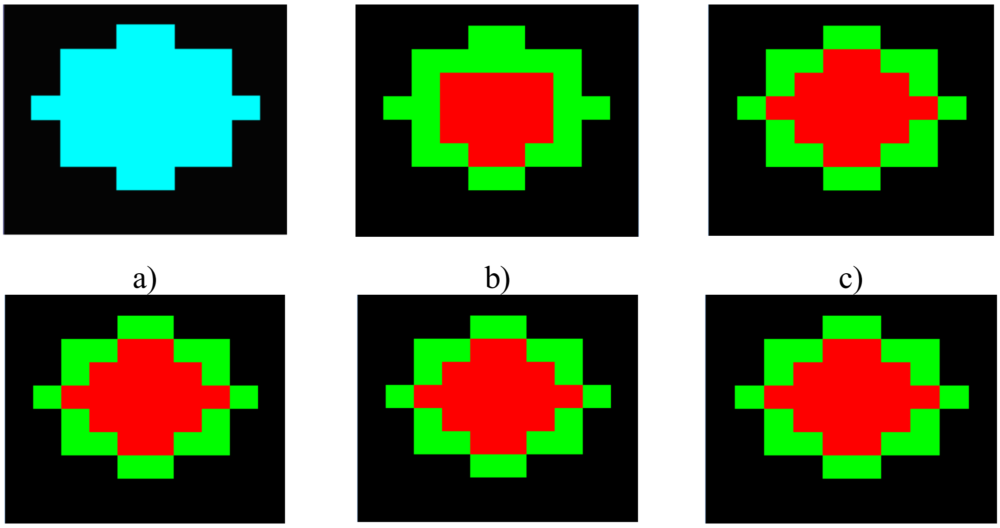
Таблиця 1 – Порівняння роботи алгоритмів проходження контуром

Алгоритм	Опис 4-зв'язного контуру	Опис 8-зв'язного контуру	Кількість перевірок, необхідних для отримання позитивного результату
Алгоритм «Жука»	+	–	1 – 4
Moore-Neighbor Tracing	+	+	1 – 7
Redial Sweep	+	+	1 – 8
Theo Pavlidi's Algorithm	+	+	(1 – 3) x 3
«Backward contour tracing»	+	+	1 – 6

Якщо є потреба отримання зв'язного контуру об'єкта, без відкидання малоінформативних пікселів, також можливе використання даного алгоритму. У такому випадку малоінформативні пікселі не відносяться до фону зображення, а залишаються в матриці контурних точок, проте їм присвоюється відповідна мітка.

Експериментальні дослідження запропонованого алгоритму

Для моделювання алгоритму «Backward contour tracing» та порівняння його з наявними було розроблено програму з використанням мови програмування Borland Delphi 7.0. Отримані результати моделювання алгоритмів наведено у відповідних таблицях. У роботі наведені результати обробки штучного об'єкта (рис. 4) та біомедичного препарату (рис. 5).



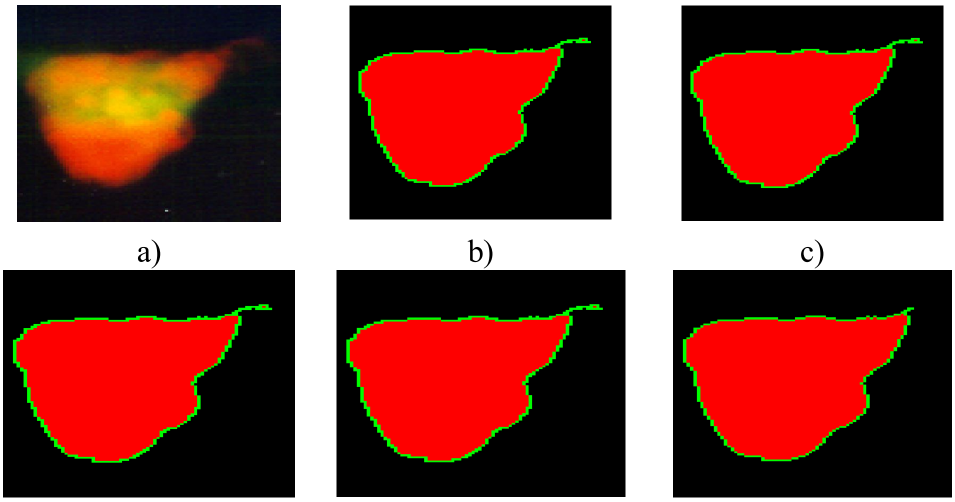
a) b) c)

d) e) f)

Рисунок 4 – Контури об’єкта: a) – вхідне зображення, b) – алгоритм «жука», c) – алгоритм Moore-Neighbor Tracing, d) – алгоритм Redial Sweep, e) – Theo Pavlidi’s Algorithm, f) – «Backward contour tracing»

Таблиця 2 – Порівняння роботи алгоритмів проходження контуром

Алгоритм	Периметр, точки	Периметр, %	Кількість перевірок	Кількість перевірок, %
Алгоритм «жука»	26	100%	55	100%
Moore-Neighbor Tracing	18	69%	76	138%
Redial Sweep	18	69%	94	170%
Theo Pavlidi’s Algorithm	18	69%	72	130%
«Backward contour tracing»	18	69%	58	106%



a) b) c)

d) e) f)

Рисунок 5 – Контури об’єкта: a) – вхідне зображення, b) – алгоритм «жука», c) – алгоритм Moore-Neighbor Tracing, d) – алгоритм Redial Sweep, e) – Theo Pavlidi’s Algorithm, f) – «Backward contour tracing»

Таблиця 3 – Порівняння роботи алгоритмів проходження контуром

Алгоритм	Периметр, точки	Периметр, %	Кількість перевірок	Кількість перевірок, %
Алгоритм «Жука»	336	100%	675	100%
Moore-Neighbor Tracing	241	72%	1052	156%
Redial Sweep	241	72%	1276	189%
Theo Pavlidi's Algorithm	241	72%	854	127%
«Backward contour tracing»	224	67%	828	123%

Дослідження проводились в рамках держбюджетної теми № 0108U002109 «Інформаційно-аналітична система для дослідження та діагностування пухлинних (ракових) клітин людини на основі аналізу їх зображень»

Висновки

Результати досліджень дозволили зробити наступні висновки щодо розглянутих та запропонованого алгоритмів:

– для проходження 8-зв'язних контурів можна використовувати усі алгоритми, окрім алгоритму «жука» (це пояснюється специфікою даного алгоритму: рух відбувається тільки в горизонтальному або вертикальному напрямках);

– найменше операцій слід виконати для алгоритму «жука». Це пояснюється нехтуванням діагональних напрямів під час проходження контуром. Перевірка додаткових пікселів не набагато збільшує кількість необхідних операцій. Винятком є контури з великою кількістю діагональних частин, тоді алгоритм «жука» значно поступається усім іншим. Для решти алгоритмів обсяг обчислень приблизно однаковий.

Алгоритм проходження контуром дозволяє відкинути малоінформативні піксели з подальшого аналізу, оскільки в алгоритмі передбачена процедура відкоту роботи. Проте коли необхідним є отримання повного зв'язного контуру об'єкта, передбачена можливість обробки малоінформативних пікселів.

Отриманий зв'язний контур об'єкта у вигляді послідовності координат точок може бути додатково проаналізований та оброблений. Наприклад, для зменшення обсягів пам'яті для зберігання контуру можна апроксимувати контур.

Література

1. Joost Vromen. Red Blood Cell Segmentation Using Guided Contour Tracing / Joost Vromen, Brendan McCane // The 18th Annual Colloquium of the Spatial Information Research Centre University of Otago. – New Zealand, 2006. – P. 65-71.
2. Fu Chang. A Component-Labeling Algorithm Using Contour Tracing Technique / Fu Chang, Chun-Jen Chen // Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003). – 2003.
3. Ramesh B.V. Zero Crossing Edge Detection and Contour Tracing for Segmentation of Cervical Cell Nucleus / B.V. Ramesh, S. Raghunandan and K.R. Ramakrishnan // Defense Science Journal. – 1993. – Vol 43, № 3 (July). – P. 223-228.
4. William K. Pratt. Digital Image Processing: PIKS Inside, Third Edition / Pratt. K. William. – New York : John Wiley and Sons, Inc., 2001 – 736 p.

5. Argyris G. 3D Terrain Modelling of the Amyntaio-Ptolemais Basin / G. Argyris, I. Kapageridis, A. Triantafyllou // 2nd International Workshop in «Geoenvironment and Geotechnics», Milos island. – Greece, 2008.
6. Contour Clustering Analysis For Building Reconstruction From Lidar Data / Jing Zhang, Lelin Li, Qiuping Lu, Wanshou Jiang // The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. – Beijing, 2008. – Vol. XXXVII. – Part B3b. – P. 135-143.
7. Гонсалес Р. Цифровая обработка зображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.
8. Сойфер В.А. Методы компьютерной обработки изображений / Сойфер В.А. – М. : Фізмат, 2003. – 784 с.
9. Ярославский Л.П. Введение в цифровую обработку зображений / Ярославский Л.П. – М. : Сов. Радио, 1979. – 312 с.
10. Березький О.М. Аналіз алгоритмів обробки зображень // Вісник Львівського національного політехнічного університету. Комп'ютерні науки та інформаційні технології / О.М. Березький, Ю.М. Батько. – 2006. – № 565. – С. 212-216.
11. Фурман Я.А. Введение в контурный анализ; приложения к обработке изображений и сигналов / Фурман Я.А., Кревецкий А.В., Передреев А.К. – М. : Физматлит, 2003. – 592 с.
12. Pavlidis T. Algorithms for Graphics and Image Processing / Pavlidis T. – Computer Science Press, Rockville, MD, 1982. – 438 p.
13. Kirk A. Dunkelberger, Robert Mitchell. Contour Tracing For Precision Measurement / A. Kirk // School of Electrical Engineering, Purdue University, West Eafayette. – Indiana, 1985. – P. 22-27.
14. Paul C. K. Kwok. Run-time Behavior of Contour Tracing Algorithms for Document Conversion / Paul C. K. Kwok, Lok C. Wong // IAPR Workshop on Machine Vision Applications. Tokyo. – Japan, 1996. – P. 361-364.
15. Din-Yuen Chan. A Contour Tracking Approach Using Progressive Boundary Marking and Synchronous Redundancy Pruning for Noisy Images / Din-Yuen Chan, Cheng-Li Chiu // Department of information Engineering, National Chiayi University, Chia-Yi. – Taiwan, 2001. – P. 261-267.
16. Crawford-Hines S. Machine-Learned Contours To Assist Boundary Tracing Tasks / S. Crawford-Hines, Ch. Anderson // Published in the Proceedings of the 1998 IEEE Southwest Symposium on Image Analysis and Interpretation. – Tucson AZ, 1998. – P. 229-231.

О.Н. Березский, Ю.М. Батько
Алгоритм прохождения контуром объекта с использованием обратного хода
В статье представлен аналитический обзор современных алгоритмов прохождения контуром, определены их преимущества и недостатки. Предложен усовершенствованный алгоритм прохождения контуром объектов с использованием обратного хода «Backward contour tracing». Разработанный алгоритм протестирован на цитологических изображениях.

O.M. Berezsky, Yu.M. Batko
Algorithm of Backward Contour Tracing
Algorithms of contour tracing are represented in the article. Their advantages and disadvantages are described. The algorithm of backward contour tracing was improved "Backward contour tracing". The developed algorithm is tested on cytological images.

Стаття надійшла до редакції 07.07.2009.